10/562705

PCT/EP2004/006532
2003P07186WOUS

IAP29 Rec'd PCT/PTO 29 DEC 2005

Description

Device and method for parameterizable controlling

The present invention relates to a control device having a plurality of inputs for respectively receiving an input real value, a plurality of outputs for respectively outputting a digital output value, a memory for storing setpoint values relating to the inputs and outputs, and an allocator for allocating a digital output value to one of the digital outputs as a function of a comparison of at least one of the input real values with a corresponding setpoint value. The present invention also relates to a corresponding method for controlling equipment.

In many applications of control technology, outputs Y_j are switched on or off as a function of inputs X_i. A control device is in this case characterized by the number of outputs j_{max} and the number of inputs i_{max}. With respectively two inputs and outputs, i.e. $j_{max} = 2$ and $i_{max} = 2$, sixteen different states are in principle conceivable. Correspondingly for controllers with eighteen input and outputs, which are widely used in control technology, more than 260,000 different states are already possible.

In equipment produced to date, all the inputs and outputs are evaluated by programmed technology. This, however, presents the following disadvantages with an increasing number of inputs and outputs (IOs): There is a great need for ROM and RAM. Furthermore, the parameterization table which increases exponentially in size requires a very large EEPROM, long reading times etc. The large number of states furthermore requires very complex parameterization and entails very long runtimes. Especially for safety technology, the latter is a great problem as regards emergency stop reaction times and maximum test times for the second fault occurrence time.

A controller of this high complexity is known, for example, from Pilz under the reference "PNOZ MULTI". A large part of the logic is in this case embodied in hardware. This is correspondingly configured extensively owing to redundancy and diversity, associated with an SFF level of more than 90% for the KAT4 safety standard. Two different controller types are in this case used with different firmware. The purpose of this is that the faster controller carries out the control functions and the slower controller is used for the monitoring.

For their part, the present Applicant sells safety equipment of the Siguard series on the market, which makes do with one firmware and one controller type, although master-slave operation is necessary in which both controllers execute all the control functions and therefore in principle require double the runtime compared with the aforementioned equipment. This disadvantage must be compensated for by a high-performance algorithm.

It is therefore an object of the present invention to provide a less elaborate controller and a corresponding method for safety technology.

According to the invention, this object is achieved by a control device having a plurality of inputs for respectively receiving an input real value, a plurality of outputs for respectively outputting a digital output value, a memory for storing setpoint values relating to the inputs and outputs, and an allocator for allocating a digital output value to one of the digital outputs as a function of a comparison of at least one of the input real values with a corresponding setpoint value, wherein an independence state value can be applied to at least one of the setpoint values in the memory, and the allocation of a digital output value

to one of the digital outputs can be carried out by the allocator independently of the at least one input real value whose allocated setpoint value has the independence state value.

The invention also relates to a method for controlling equipment by receiving a plurality of input real values, providing setpoint values relating to inputs and outputs, establishing a digital output value as a function of a comparison of at least one of the input real values with a corresponding one of the setpoint values, outputting the digital output value, applying an independence state value to at least one of the setpoint values, and establishing the digital output value independently of the at least one input real value whose allocated setpoint value has the independence state value.

In safety technology, the error susceptibility and verifiability of the algorithm are of prime importance. If the computing outlay is reduced according to the invention, a reliable control function can therefore be readily achieved in master-slave operation.

The control device according to the invention may comprise a first evaluator for converting input raw values into digital input values for the further processing as input real values. This makes it possible, for example, to classify analog input signals as an active or inactive input.

A second evaluator may furthermore be provided in the control device, which is connected downstream of the first evaluator. This allows the digital input values to be allocated to logical input states for the further processing as input real values.

Preferably, the setpoint values respectively have one of the state values 1, 0 and independence state value. In this way,

for example, it is possible to produce the binary states "TRUE" and "FALSE" as well as a state which is insignificant for the output result.

A plurality of sets of setpoint values are preferably stored respectively for an output value or set of output values in the memory. In this way, a plurality of parameterizations can be stored simultaneously in the equipment.

The control device according to the invention may have a safety instrument by which the equipment to be controlled can be switched to a safety state. For example, it may be switched to the safety state if the output real values deviate from the corresponding setpoint values for more than a predetermined time. In a special example of this, the control device may comprise two controllers which both execute the algorithm and store all fulfilled parameterizations as well as the output vector Y_j in binary form. These stored values are compared in each cycle. If they deviate for a time which is longer than a predetermined maximum time, then the equipment to be controlled is switched to a safe state.

The safety device may be optimized by checking the sets of setpoint values with a check sum at fixed time intervals. In particular, a setpoint value matrix i.e. a fixed parameterization, which is stored in the memory, may be secured by a cyclic CRC (cyclic redundancy check sum) and verified at fixed time intervals in order to discover errors in the matrix S or in the memory. In this way, a variable function can be checked for errors straightforwardly.

The present invention will now be explained in more detail with the aid of the appended drawings, in which:

FIG 1 shows an outline flow chart of the preprocessing of the input real values; and

FIG 2 shows a logic diagram for the allocation of output states according to the invention.

The exemplary embodiments described in detail below represent preferred embodiments of the present invention.

The outputs Y of the control-safety equipment are the result of a switching function H with input X:

$$Y = H(X)$$

The input X, or the plurality of inputs X_i, in this case may respectively have the following states independently of its/their function:

		0 ("FALSE")	input must be inactive
X_i	=	1 ("TRUE")	input must be active
		D ("DONTCARE")	input state may be anything

In control technology, an active output state Y_j is generally reached for precisely one or very few input state vectors. For most of the input state vectors X_i, the output or outputs are inactively configured. With uncorrelated inputs, i.e. inputs that do not affect one another, for example operating selector switch, muting, key switch or the like, there are usually at most j_{max} input state vectors for j_{max} active outputs Y_j.

If the inputs are correlated, however, then:

Number of the active output states $Y_j < \sum_{1}^{j_{max}} (\prod Z_i)$

Here, Z_i corresponds to the number of correlations of the inputs X_i. In the limiting case of uncorrelated inputs, $Z = 1$ since the inputs are then only correlated with themselves.

The evaluation of the inputs is carried out in two stages according to this example according to the invention, as indicated in FIG 1. Raw input data R_i, for example analog signals or digital signals of any level, are first subjected to a physical evaluation. Here, for example, the allocation $X_i = 1$ is made when the corresponding input is active, and $X_i = 0$ when the input is inactive.

In a second step S2, the digital input values X_i are logically evaluated. Each input therefore has a function ID, for example ID_1 = ID_POWERBUTTON. A logical input state or function value F_i is assigned to each digital input value X_i. In the example, $F_i = 1$ would apply if the power button has been actuated successfully, and $F_i = 0$ would apply if the power button has not been actuated or has not been actuated successfully.

A logical allocation is carried out in the further step S3, each real value F_i being compared with a setpoint value S_i. A corresponding output value Y_j results from this comparison. Preferably, the controller is configured so that n_{max} different parameterizations can be stored in it. This means that for all the n_{max} parameterizations, a set of setpoint values $S_{i,n}$ is respectively stored. They have the values

	0 ("FALSE")	input must be inactive
$S_{i,n}$	1 ("TRUE")	input must be active
	D ("DONTCARE")	input state may be anything

FIG 2 shows a flow chart for determining the output states Y_j. In an initialization step S4, the number of the parameter set is put at n = 1 and the output value Y_j is put at zero. In a further step S5, the logical input states F_i for each parameterization n are compared with the allocated threshold value $S_{i,n}$ (comparison operator "=="). All the comparisons are combined by the AND operator "&&". If the overall result of the comparisons is "TRUE", then the respective output Y_j receives the value of the logic operation "Y_j OR $Y_{j,n}$". In this case, $Y_{j,n}$ corresponds to the value stored as a setpoint value together with $S_{i,n}$.

The comparison routine of step S5 is repeated n times according to step S6. After this, the output value assignment is ended according to step S7.

For each parameterization, the output Y_j with $Y_{j,n} = 1$ may then be connected up or activated. Otherwise, the respective output Y_j is inactive.

According to the invention, not every real value F_i is compared with the corresponding setpoint value $S_{i,n}$ in step S5. Rather, a comparison is only carried out if the setpoint value $S_{i,n}$ does not have the value "D". This can avoid a multiplicity of comparison operations. The total runtime for determining the output states is correspondingly reduced.

If the inputs are mutually independent, for example in the case of parallel switches, then the number of parameterizations n_{max} is equal to the total number of outputs j_{max}. If the inputs are dependent on one another, however, for example switches connected in series, then two parameterizations may for example be necessary for one output.

In a specific example, eleven independent inputs are applied to the controller in order to control four outputs.

Accordingly, four different parameterizations must be stored in the controller.